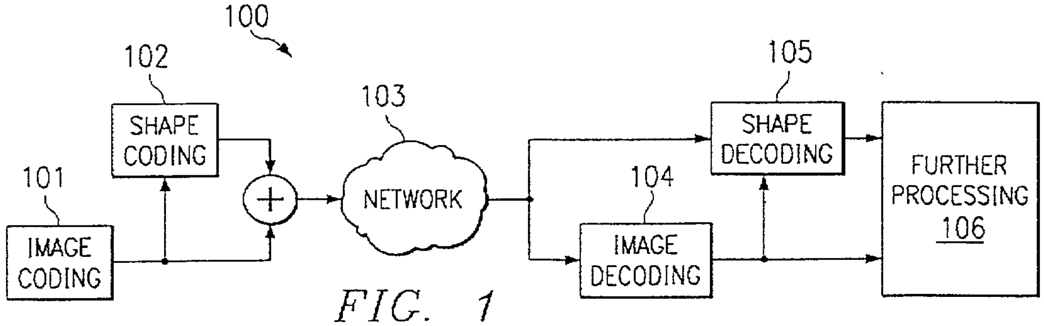

100
102
SHAPE CODING
101
IMAGE CODING
103
NETWORK
105
SHAPE DECODING
104
IMAGE DECODING
FURTHER PROCESSING
106

FIG. 1

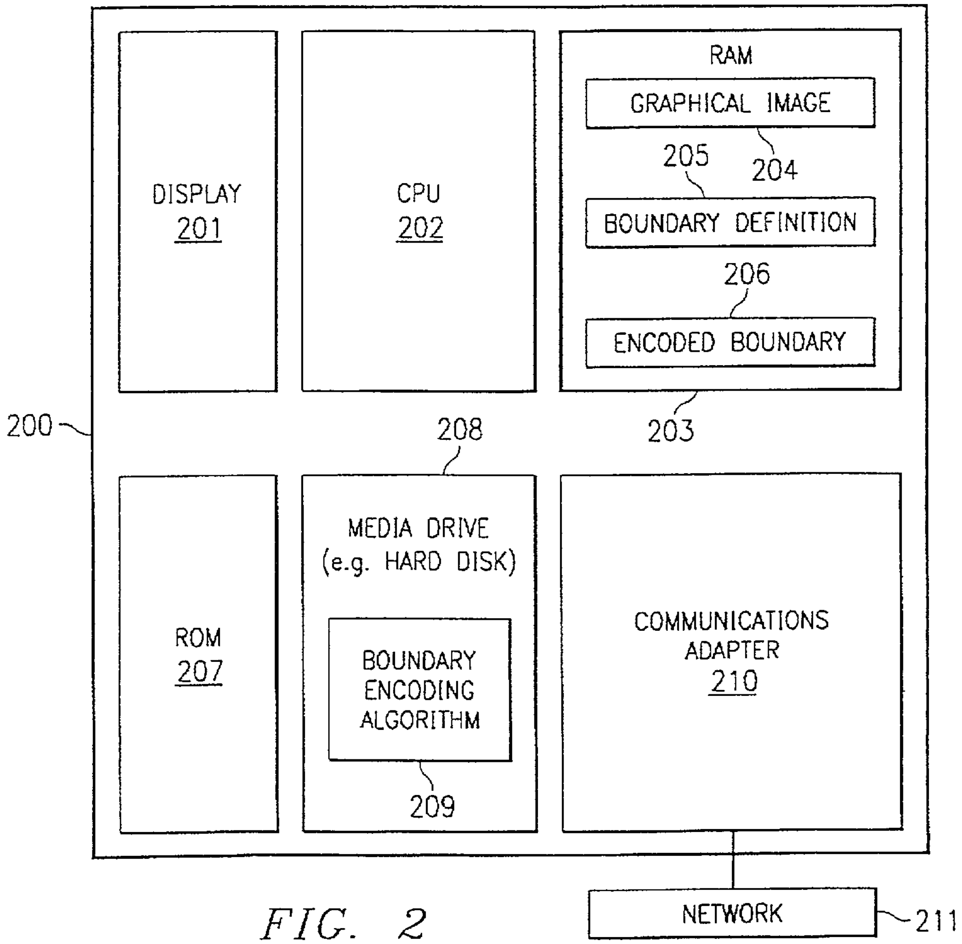

DISPLAY
201
CPU
202
RAM
GRAPHICAL IMAGE
205
204
BOUNDARY DEFINITION
206
ENCODED BOUNDARY
200
208
203
ROM
207
MEDIA DRIVE
(e.g. HARD DISK)
BOUNDARY ENCODING ALGORITHM
209
COMMUNICATIONS ADAPTER
210
NETWORK
211

FIG. 2

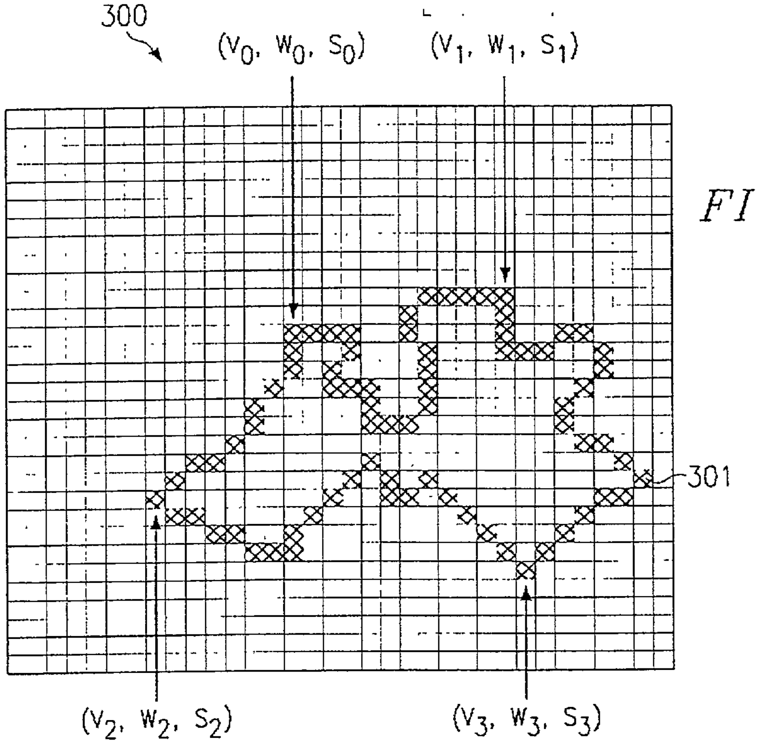


FIG. 3

FIG. 4

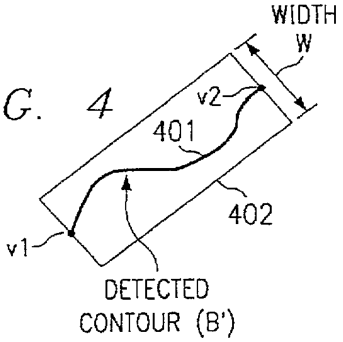


FIG. 5

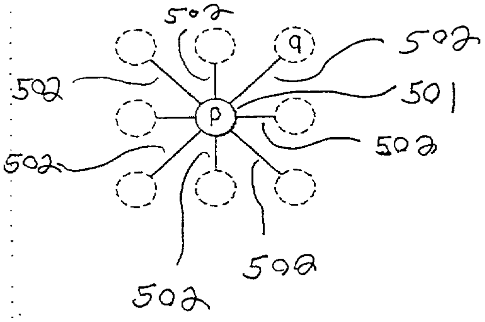


FIG. 11

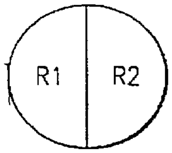


FIG. 6

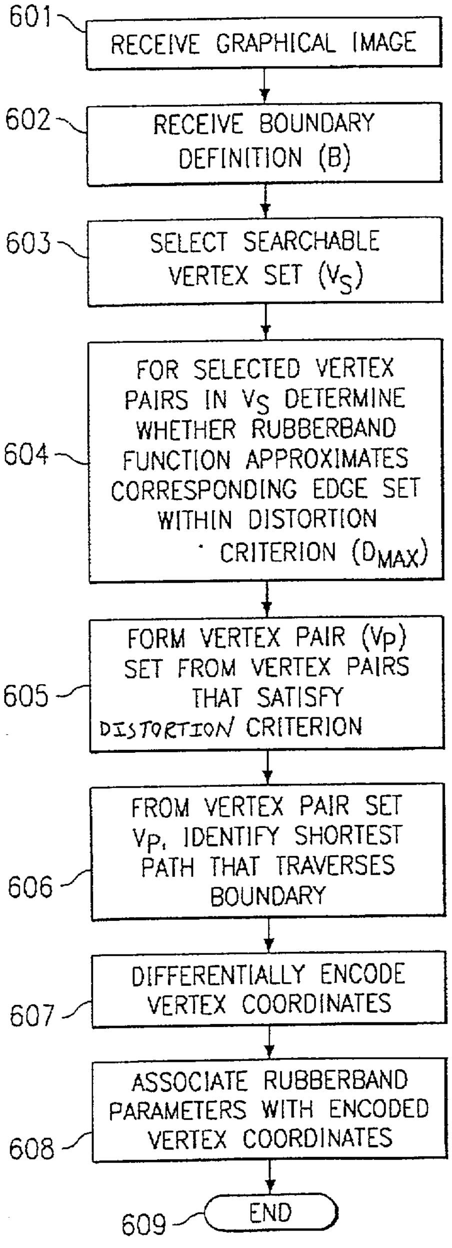

601
RECEIVE GRAPHICAL IMAGE
602
RECEIVE BOUNDARY DEFINITION (B)
603
SELECT SEARCHABLE VERTEX SET (V_S)
604
FOR SELECTED VERTEX PAIRS IN V_S DETERMINE WHETHER RUBBERBAND FUNCTION APPROXIMATES CORRESPONDING EDGE SET WITHIN DISTORTION CRITERION (D_{MAX})
605
FORM VERTEX PAIR (V_P) SET FROM VERTEX PAIRS THAT SATISFY DISTORTION CRITERION
606
FROM VERTEX PAIR SET V_P, IDENTIFY SHORTEST PATH THAT TRAVERSES BOUNDARY
607
DIFFERENTIALLY ENCODE VERTEX COORDINATES
608
ASSOCIATE RUBBERBAND PARAMETERS WITH ENCODED VERTEX COORDINATES
609
END

FIG. 7

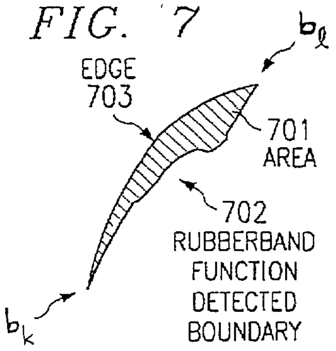


FIG. 8

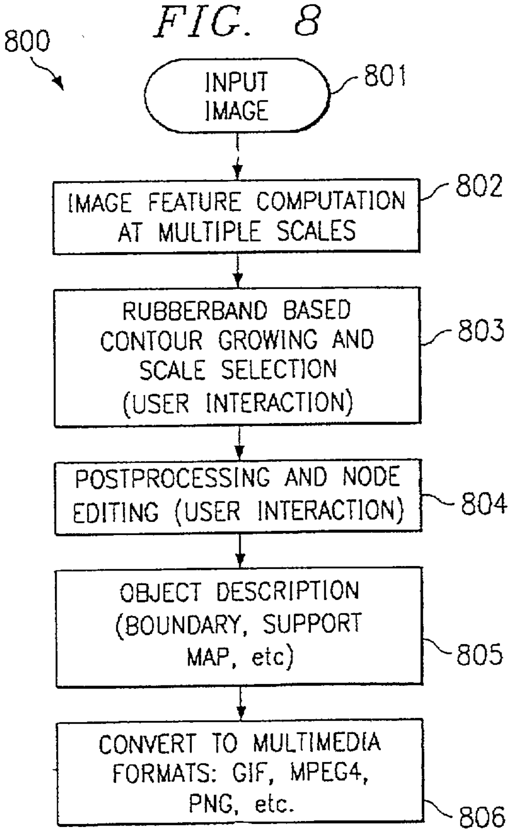

800
INPUT IMAGE
801
IMAGE FEATURE COMPUTATION AT MULTIPLE SCALES
802
RUBBERBAND BASED CONTOUR GROWING AND SCALE SELECTION (USER INTERACTION)
803
POSTPROCESSING AND NODE EDITING (USER INTERACTION)
804
OBJECT DESCRIPTION (BOUNDARY, SUPPORT MAP, etc)
805
CONVERT TO MULTIMEDIA FORMATS: GIF, MPEG4, PNG, etc.
806

FIG. 9A

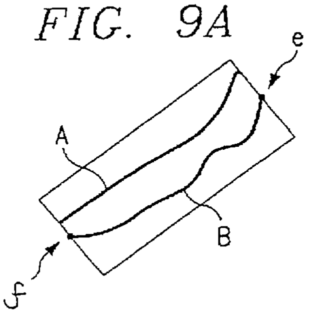


FIG. 9B

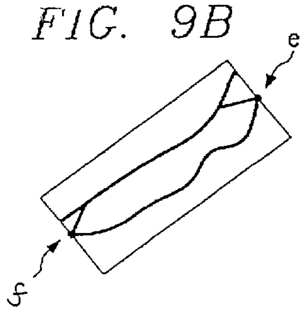
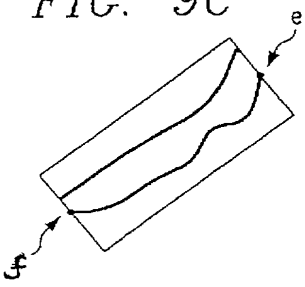


FIG. 9C



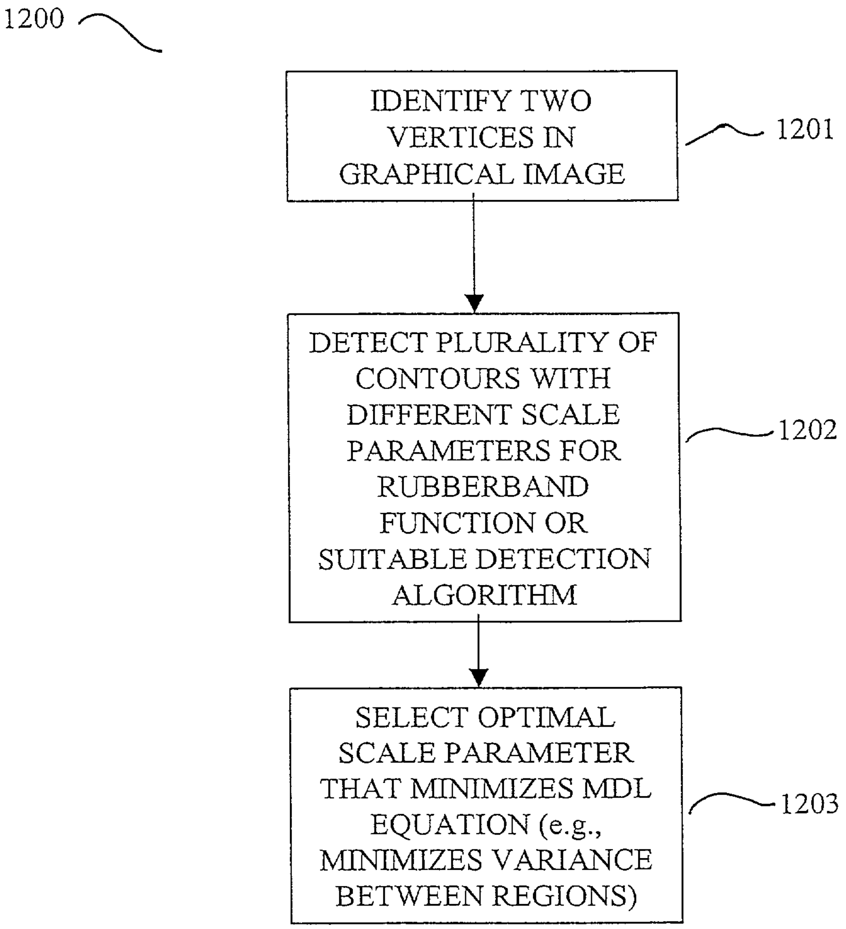
4 7607 - P523US - 10111466

5/6

FIG. 10

```
Input: f (start point), e (ending point), Dist(p,q) (local distance
    definition)
Assistant Data Structure:
  L1 (active list 1)
  L2 (active list 2)
  C(p) (cumulative distance from f to p)
Output: ptr (minimal cost path pointers)
Algorithm:
(1001)  Initialize assistant data structure (L1, L2 are set empty, and
        C to $+\infty$).
(1002)  Set initial threshold $T_0$ and increasing step $\delta_T$.
(1003)  $T = T_0$;
(1004)  push(L1, f, 0 );
(1005)  while( $T <= T_{\max}$ and C(e)==+$\infty$ ) {
(1006)      while( num(L1)>0 ) {
(1007)          pop( L1, p );
(1008)          flag_thresholded=0;
(1009)          for (each $q \in N(p)$) {
(1010)              if( Dist(p, q)>T ) {
(1011)                  flag_threshold=1; continue;
(1012)              }
(1013)              $d' = C(p) + \mathrm{Dist}(p, q)$;
(1014)              if( $d'$ <C(q) ) {
(1015)                  if( q is in L1 ) remove( L1, q );
(1016)                  C(q)=$d'$;
(1017)                  ptr(q)=p;
(1018)                  push( L1, q, $d'$ );
(1019)              }
(1020)          } //end of for
(1021)          if( flag_threshold ) {
(1022)              push( L2, p, C(p) );
(1023)          }
(1024)      } //end of inner while
(1025)      T=T+$\delta_T$;
(1026)      Copy L1 from L2 and clean L2.
(1027)  } //end of outer while
```

FIG. 12


1200
IDENTIFY TWO VERTICES IN GRAPHICAL IMAGE
1201
DETECT PLURALITY OF CONTOURS WITH DIFFERENT SCALE PARAMETERS FOR RUBBERBAND FUNCTION OR SUITABLE DETECTION ALGORITHM
1202
SELECT OPTIMAL SCALE PARAMETER THAT MINIMIZES MDL EQUATION (e.g., MINIMIZES VARIANCE BETWEEN REGIONS)
1203